PRENTICE COMPUTER CENTRE

UNIVERSITY OF QUEENSLAND, ST. LUCIA, QUEENSLAND, AUSTRALIA. 4067.

NEWSLETTER

N-261   30-March-81

CONTENTS

Authorized by the Director of the Computer Centre

Principal Service Centres

1.0 NEW VERSION OF SPSS

The latest release of SPSS, version 8.0, has arrived and been installed. At present this only applies to the KL, but it should be mounted shortly on the KA.

This new release is compatible with our current version 7, except that workspace requirements may have changed some procedures.

The principal enhancements of this release are two new procedures - REPORT and SURVIVAL.

REPORT can be used to produce listings of data in user specified formats. At the users option are such things as headings, column layouts, subtotals and summary statistics. With its statistical capabilities REPORT can even be used to produce the counts, means etc. required without recourse to procedures such as FREQUENCIES.

SURVIVAL evaluates the time interval between two events, a starting event and a terminal event. Though most commonly associated with medical studies, it could, for example, be used in the social sciences to describe and compare the time interval from marriage to divorce for groups of people with different social backgrounds.

There are also numerous other minor improvements to existing procedures. See the new manual for details.

The documentation for the new procedures is the small blue covered manual 'SPSS Update'. This is a supplement to the existing maroon covered one. Copies of 'SPSS Update' will be available shortly from the batch station.

Chris McGovern
extension 3391

2.0 1022 DATA BASE MANAGEMENT SYSTEM

The latest release of 1022 has been put on NEW: for user testing. Barring any problems, it is proposed to make this the standard version on Monday, 11 May 1981. Users of 1022 are advised to try out the new release prior to this date and notify the Computer Centre of any problems.

To access it use

.R NEW:1022   (interactively)
and

NEW:HR1022/SEA etc.   (host loading the language interface)

Though this is mainly a maintenance release, the following enhancements are noteworthy.

New date functions to return the elements of a date, the day of the week, the number of weekdays between two dates, etc.

Two new text functions $TEXTL and $TEXTR which left and right justify text attributes with truncation if necessary. A use of these functions is to extract an arbitrary substring from a given string. For example, suppose attribute A contains 'ABCDEF` and we wish to extract the string 'BCD` and put it in C then

```
DEF TEXT 4 B TEXT 3 C TEXT 6 A
LET B EQ $TEXTL A.
LET C EQ $TEXTR B.
```

Note that the first LET statement could simply be

```
LET B EQ A.
```

as the different field lengths would force the truncation. $TEXTL is used purely for clarity in the example.

The new relational operator BEGINS allows searches for an attribute value that begins with a given text string. The operator BETWEEN allows a simple specification of a search for an attribute value in a certain range.

The documentation for this version are the files MANUAL.MEM and HOST.MEM also on NEW:.

Chris McGovern
extension 3391

3.0 CHANGES TO PASCAL FOR 7.01

Two changes have been made to the PASCAL compiler since the 7.01 monitor was installed. Firstly, the compiler made the assumption that two buffers would always be allocated when doing file I/O, however, the 7.01 monitor had a default allocation of six buffers which caused some problems with programs which previously ran successfully. The buffer allocation has been changed to always use two buffers regardless of any default buffer setting the user has previously made. Programs which were compiled prior to 7.01 will not generally run under 7.01 and should be recompiled.

The other change was to remove the old KA floating point instructions from the SIN and COS routines on the KL. People with programs compiled before 7.01 who use these routines will need to do one of:-

1. Recompile the program, or

2. SET FLOATING POINT (see N258) before running the programs.

> Mark Robbie
> extension 3022

4.0 NEW VERSION OF DDT

DDT is the Dynamic Debugging Technique software for the DEC-10 which allows users to quickly and easily locate errors in their programs. The users program is loaded with the DDT package, and when run, DDT allows the user to temporarily halt the execution of his program at any point, inspect the contents of any location (code or data variable) at that point, modify the contents of any location and then continue execution to the next defined break (or halt) point. In this way, a user can 'step' through his program to find exactly where it starts going wrong.

The Centre is conducting an introductory course on FORDDT (DDT for Fortran Programs) on 26-27 May (2pm - 5pm) and interested users should enrol for this.

The following article which describes some of the features of a new version of DDT available on the directory NEW:, is intended for the experienced MACRO programmer and DDT user.

This new version (41) contains many new features and many minor enhancements, however, SOME OF THE CHANGES MAY CAUSE COMMANDS THAT WORKED UNDER PREVIOUS VERSIONS TO NOT WORK OR TO WORK DIFFERENTLY.

The incompatibilities are due to the fact that DDT now handles 36-bit addresses. The changes are listed in DOC:DDT.DOC, with fuller explanations in DOC:DDT41.MAN.

Briefly, all addresses used by DDT are now 36-bit numbers, including the current location counter ".". Commands like "/" that take addresses as arguments will now take all 36 bits of the argument as the address to be used. This change can cause DDT to type "?" unexpectedly on some commands that worked with previous versions of DDT. The following typescript illustrates a typical sequence that performs differently with DDT 41:

```
FOO/    <old contents>    -1,,TEXT<tab>
777777,,TEXT/    ?
777777,,TEXT+1/    ?
```

The user obviously expected to open location TEXT, but DDT 41 interpreted the entire expression as an address and opened a very large nonexistent address. Another example is:

```
DISP/ 10327,,17346 $$Q/ ?   .=17346,,10327
```

In this case, the user wanted to follow the address in the left half of location DISP, but ended up looking at a huge address instead. The $$Q means "the last quantity typed out with the halves swapped", which has non-zero bits in the left half. The .= verifies that this is indeed what has happened.

If DDT types "?" when it should have typed real data, and you suspect that you may have ended up at a huge address, you can type .= or linefeed to verify whether or not this has indeed happened. If it has, the command ,./ (three adjacent keys on most terminals) will quickly get you to the corresponding address with a zero left half. You can get DDT to ignore the left half of the argument to a location opening command such as / or tab by including a space somewhere in the argument. Space means to retain the left half of the current location when opening a new location. For instance, the first example above could have been done as follows:

```
FOO/    <old contents>    -1,,TEXT<space><tab>
TEXT/    ANDI 360732(16)
TEXT+1/    CONSZ 30,500000(14)
```

This command sequence would deposit -1,,TEXT in location FOO, but open location 0,,TEXT, since the left half of the current address "." is 0. Another example is:

```
LOOP1+3/    PUSHJ P,ERROR    JRST PAT..<tab>
PAT../    0
```

In this example, the user wanted to deposit a JRST PAT.. at LOOP1+3, and then open location PAT.. to write a patch. However, the user did not have to include an explicit space before the tab, since

4

the argument to the tab already contained the space between the JRST and the PAT... In fact, this kind of example is exactly why space has been implemented as a "retain the current section" command. The ramifications of 36-bit addresses are more complex than can be described in this short summary. Refer to the file DDT41.MAN for complete documentation.

Auto-typeout On Breakpoints:-

The command to set a location to type out automatically on a breakpoint has changed, as follows:

Old way:
A,,B$B or B(A)$B

New way:
A<B$B

This command means to set a breakpoint at location B, and type out location A symbolically every time the program hits the breakpoint. If the old-style command is typed to DDT 41, it will be interpreted as a command to set a breakpoint at the full-word address A,,B, which is usually illegal.

The new version will be moved to SYS: at the end of April.

Will Gout
extension 3023

5.0 SYSTEMS PERFORMANCE - FEBRUARY

KA10

One of the most frequently occurring problems on the KA System is that of intermittent memory parity errors and this was the cause of the February downtime.

KL10

The KL10 also had intermittent faults during February which were caused by a fault in a DTE which links the KL to one of the PDP11/40 Communications Processors. In addition, we believe that dirty contacts on the clock board caused a number of the unexplained crashes.

Graham Rees
extension 3288

S Y S T E M   P E R F O R M A N C E   R E P O R T

For node   KA10   there were   27 working days in the period   1/Feb/81 to 28/Feb/81

		<	KA10	>
			HHH:MM	%
1.	Attended system running time		447:50	
2.	Plus unattended system running time		167:28	
3.	Equals total system running time		615:18	100.0
	less time used for:			
4.	Scheduled maintenance		18:39	3.0
5.	Dedicated operations tasks		3:11	0.5
6.	Dedicated systems development		0:00	0.0
7.	Equals time scheduled for use		593:28	96.5
	less lost time due to:			
8.	Unscheduled maintenance		0:30	0.1
9.	Hardware faults		7:11	1.2
10.	Software faults		0:00	0.0
11.	Unresolved		2:54	0.5
12.	Environmental conditions		0:09	0.0
13.	Equals time available to users		582:44	94.7
14.	Effective user uptime (13./7.)			98.2
15.	Number of crashes			11
16.	Mean availability between crashes			52:59
17.	Mean time to recover crashes (minutes)			55
18.	Total number of Jobs			1343

SYSTEM PERFORMANCE REPORT

For node KL10   there were 26 working days in the period 1/Feb/81 to 28/Feb/81

		< KL10 >		< DN87A >		< DN87B >	
		HHH:MM	%	HHH:MM	%	HHH:MM	%
1.	Attended system running time	457:31		457:31		457:31	
2.	Plus unattended system running time	129:15		129:15		129:15	
3.	Equals total system running time	586:46	100.0	586:46	100.0	586:46	100.0
	less time used for:						
4.	Scheduled maintenance	20:17	3.5	20:17	3.5	20:17	3.5
5.	Dedicated operations tasks	9:23	1.6	9:23	1.6	9:23	1.6
6.	Dedicated systems development	7:06	1.2	7:06	1.2	7:06	1.2
7.	Equals time scheduled for use	550:00	93.7	550:00	93.7	550:00	93.7
	less lost time due to:						
8.	Unscheduled maintenance	2:06	0.4	0:00	0.0	0:00	0.0
9.	Hardware faults	6:54	1.2	0:04	0.0	0:00	0.0
10.	Software faults	0:33	0.1	0:27	0.1	0:00	0.0
11.	Unresolved	2:29	0.4	1:24	0.2	0:19	0.1
12.	Environmental conditions	0:37	0.1	0:37	0.1	0:37	0.1
13.	Equals time available to users	537:21	91.6	547:28	93.3	549:04	93.6
14.	Effective user uptime (13./7.)		97.7		99.5		99.8
15.	Number of crashes		28		33		4
16.	Mean availability between crashes		19:11		16:35		137:16
17.	Mean time to recover crashes (minutes)		21		3		5
18.	Total number of Jobs		11242				

7.

```
S Y S T E M    P E R F O R M A N C E    R E P O R T

For node GRIFFITH there were  27 working days in the period  1/Feb/81 to 28/Feb/81


                                              <     GRIFFITH    >

                                              HHH:MM          %

   1.  Attended system running time          157:53

   2.  Plus unattended system running time   411:53

   3.  Equals total system running time      569:46       100.0

            less time used for:

   4.        Scheduled maintenance              0:00         0.0

   5.        Dedicated operations tasks         0:00         0.0

   6.        Dedicated systems development      0:00         0.0

   7.  Equals time scheduled for use         569:46       100.0

            less lost time due to:

   8.        Unscheduled maintenance            0:00         0.0

   9.        Hardware faults                    2:44         0.5

  10.        Software faults                    0:00         0.0

  11.        Unresolved                         1:03         0.2

  12.        Environmental conditions           0:29         0.1

  13.  Equals time available to users        565:30        99.3

  14.  Effective user uptime (13./7.)                       99.3


  15.        Number of crashes                               17

  16.        Mean availability between crashes            33:16

  17.        Mean time to recover crashes (minutes)          13
```

8.

For node COMMERCE there were 20 working days in the period 1/Feb/81 to 28/Feb/81

		< COMMERCE >	
		HHH:MM	%
1.	Attended system running time	163:02	
2.	Plus unattended system running time	0:00	
3.	Equals total system running time	163:02	100.0
	less time used for:		
4.	Scheduled maintenance	0:00	0.0
5.	Dedicated operations tasks	0:00	0.0
6.	Dedicated systems development	0:00	0.0
7.	Equals time scheduled for use	163:02	100.0
	less lost time due to:		
8.	Unscheduled maintenance	0:00	0.0
9.	Hardware faults	28:30	17.5
10.	Software faults	0:00	0.0
11.	Unresolved	1:29	0.9
12.	Environmental conditions	0:00	0.0
13.	Equals time available to users	133:03	81.6
14.	Effective user uptime (13./7.)		81.6
15.	Number of crashes		13
16.	Mean availability between crashes		10:14
17.	Mean time to recover crashes (minutes)		138

9.